

Gamesino Functional Specification

South East Technological University



Student: *Samuel David*

Supervisor: *Greg Doyle*

Submission Date : *17/04/2023*

Table of Contents

Introduction	1
Use Case Diagrams	2
CRUD Account (Create)	2
Login	4
Browse Store	5
Purchase Game	5
View Library	6
Play game	7
View Payments	8
Logout	9
FURPS	9
Metrics / Success	11

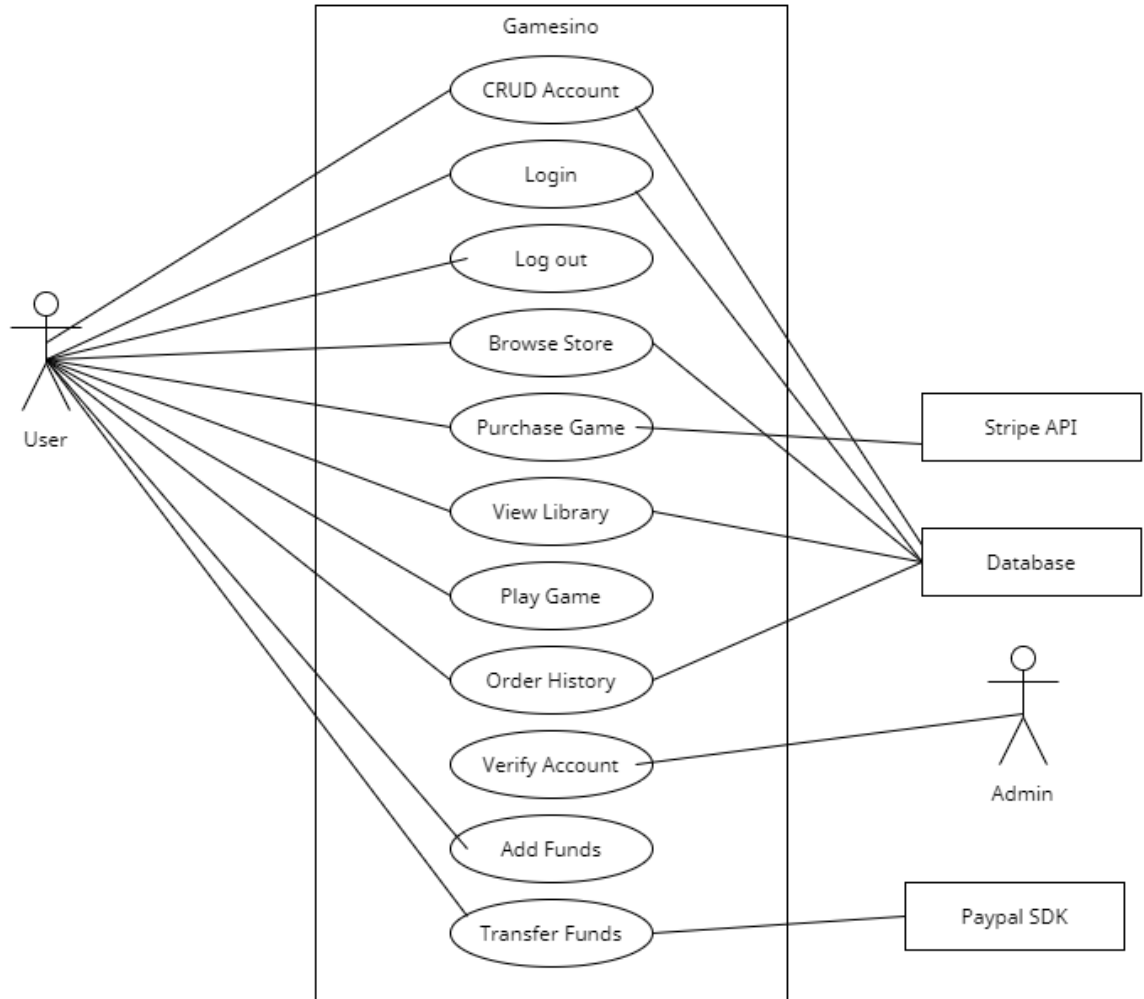
Introduction

The following research document will provide an insight into the development of this project.

The goal of this project is to develop an online games platform application that allows people to purchase/play online games or embedded web games from their Library. Users will also be able to gamble in a range of casino games to win some cash.

In this project, it will go into great detail what were the requirements of this project, already available Gaming platforms that has the same/similar concept to this project, how the games are being stored, the security that's implemented into the project, the different payment systems that can be used and discuss the numerous types of languages and frameworks that went into the making of the application.

Use Case Diagrams



CRUD Account (Create)

Use Case Name	CRUD ACCOUNT
Actors	User, Database
Brief Description	This use case explains a user creating an account on the application
Main Success Scenario	<ol style="list-style-type: none"> 1. The User enters a username 2. The User enters their email address 3. The User enters a password 4. The User confirms their password
Alternatives	<ol style="list-style-type: none"> 1a. The User's username is already taken <ol style="list-style-type: none"> 1. The user is prompted to enter a new username or use a suggested username 2a. The user enters an invalid email address <ol style="list-style-type: none"> 1. The user is prompted to enter a new email address 3a. The user enters a password that doesn't meet requirements <ol style="list-style-type: none"> 1. The user is prompted to enter a new password that meets requirements 4a. The user's password doesn't match it's "confirm password" <ol style="list-style-type: none"> 1. The user is prompted to make sure both passwords match

Login

Use Case Name	Login
Actors	User, Database
Brief Description	This use case explains a user login into a account on the application
Main Success Scenario	<ol style="list-style-type: none"> 1. The user enters their username 2. The user enters their password

	<ol style="list-style-type: none"> 3. The user is asked to confirm an login email 4. The user submits their information
Alternatives	<ol style="list-style-type: none"> 1a. The user enters the wrong username <ol style="list-style-type: none"> 1. The user is prompted to enter the correct username 2a. The user enters the wrong password <ol style="list-style-type: none"> 1. The user is prompted to enter an valid password 2. The user can reset their password using "Forget your password"

Browse Store

Use Case Name	Browse Store
Actors	User, database
Brief Description	This use case explains a user browsing the store on the application
Main Success Scenario	<ol style="list-style-type: none"> 1. The user login 2. Gamesino opens up store page 3. The user can browse for games using Games tab to display all games 4. The user can search for games using search bar on the application 5. The user can pick a category "Action" full of games
Alternatives	<ol style="list-style-type: none"> 4a. The user searched for a game that doesn't exist <ol style="list-style-type: none"> 1. The application will say game doesn't exist and display other options

Purchase Game

Use Case Name	Purchase Game
Actors	User, Stripe

Brief Description	This use case explains a user purchasing a game on the application
Main Success Scenario	<ol style="list-style-type: none"> 1. The use login 2. The user browses the store 3. The user selects a game 4. The user adds game to cart 5. The user proceeds with checkout 6. The games application displays their checkout form powered by payment provider 7. The user enters their card number 8. The user enters in their card expiry date 9. The user enters in billing address 10. The user enter submits form to payment provider 11. The payment provider takes payment from card issuer 12. The game application displays order confirmation 13. The games application adds purchased game to user's library
Alternatives	<p>7a. The card number is not valid</p> <ol style="list-style-type: none"> 1. The form prompts user to enter valid card details <p>8a. The user enters an expired card date</p> <ol style="list-style-type: none"> 1. The form prompts user to enter a valid expiry date <p>10a. The user forgets to enter in a detail in the form</p> <ol style="list-style-type: none"> 1. The form redirects user to missing field. <p>11a. The payment provider is unable to receive funds.</p> <ol style="list-style-type: none"> 1. The payment provider cancels payment 2. The form prompts user to informs user of unable to receive payment from card 3. The user is redirected to checkout form

View Library

Use Case Name	View Libaray
Actors	User, database
Brief Description	This use case explains a user viewing their library of games on the application
Main Success Scenario	<ol style="list-style-type: none"> 1. The user login 2. The user goes to their game library 3. Gamesino displays all their own games in their Library
Alternatives	<p>2a. The user doesnt have any games</p> <ol style="list-style-type: none"> 1. The games platform will prompt user to buy some games from the store.

Play game

Use Case Name	Play game
Actors	User, database
Brief Description	This use case explains a user playing a game from their library
Main Success Scenario	<ol style="list-style-type: none"> 4. The user login 5. The user goes to their game library 6. Gamesino displays all their own games in the library page 7. The user selects the game of their choice 8. The user clicks play button to play game 9. Gamesino redirects the user to their game via an URL 10. If it's a embedded browser game,

	game will launch in browser
Alternatives	2a. The user doesn't have any games in their game library to play <ol style="list-style-type: none"> 1. Gamesino prompts user to buy games to add the game library

Order History

Use Case Name	Login
Actors	User, Database
Brief Description	This use case explains a user viewing their recent transactions/payments made to the games platform
Main Success Scenario	<ol style="list-style-type: none"> 1. The user logs in 2. The user presses Order History Tab from their account 3. The games application displays their recent transactions
Alternatives	3a. The user has made no purchases to the games application <ol style="list-style-type: none"> 1. The game application will display text saying there is no purchases

Verify Account

Use Case Name	Login
Actors	User, Admin
Brief Description	This use case explains an admin user approving/declining a user's verification request
Main Success Scenario	<ol style="list-style-type: none"> 1. The admin logs in. 2. Admin opens admin tab on gamesino website 3. Admin opens verification requests tab 4. Admin selects which user to update verification status 5. Admin selects approve or decline for

	verification status. 6. Admin updates user status
Alternatives	3a. No pending requests displayed 1. Gamesino displays a message to the admin, there is no requests yet. 5a. User is already approved or declined 1. Admin logs out

Add Funds

Use Case Name	Login
Actors	User, Stripe
Brief Description	This use case explains an user adding funds to their balance using stripe
Main Success Scenario	<ol style="list-style-type: none"> 1. The user logs in 2. The user clicks on Add funds tab 3. The user chooses which fund level they wish to add to account 4. User enters in their details using stripe API 5. The stripe API accepts the payment. 6. User is redirected to success page 7. Funds are added to users balance
Alternatives	5a. Stripe declines payment 1. User is prompted by stripe that their payment has been unsuccessful. 4a. User backs out of Stripe API 1. User is redirected to Cancel Page

Logout

Use Case Name	Login
Actors	User, Database

Brief Description	This use case explains a user logging out of the system.
Main Success Scenario	7.
Alternatives	<ol style="list-style-type: none"> 1a. The user enters the wrong username 2. The user is prompted to enter the correct username 2a. The user enters the wrong password 3. The user is prompted to enter an valid password 4. The user can reset their password using "Forget your password"

Transfer Funds

Use Case Name	Login
Actors	User, PayPal API
Brief Description	This use case explains a user transferring their funds to PayPal
Main Success Scenario	<ol style="list-style-type: none"> 1. The user logs in 2. The user clicks on casino Tab 3. The user selects a casino game 4. The user wins the casino game 5. The user gets their won amont added to their balance 6. The user clicks on add funds tab 7. The user enters in details in paypal 8. Funds transferred successfully
Alternatives	<ol style="list-style-type: none"> 2a. User has Verification checked <ol style="list-style-type: none"> 1. User gets redirected to verify account page 2. User's verification has been declined. 3a. No balance to pay casino Game <ol style="list-style-type: none"> 1. Gamesino displays a prompt telling the user that there is no funds to play

	<p>game and has a option to go to add funds page</p> <p>7a. User enters in the wrong details for the paypal</p> <p>8a. Insufficient funds</p> <ol style="list-style-type: none"> 1. User gets told they dont have enough funds to transfer,
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

FURPS

FURPS stands for Functionality, Usability, Reliability and lastly performance. This topic will go into detail on these sub headings of the project. This is useful to the project as functional aspects and non functional aspects of the project will be discussed and many others.

Functionality

The functionality in FURPS represents the main use cases of the application. Gamesino is a Online games dashboard/ E-commerce store. Being able to buy and play games would be one of the main functionalites of this application.

Main functionalites are:

- Secure Delivery of games
- Secure storage of games and game states
- Payment of winnings
- Transaction history
- And more

Usability

The usability in FURPS represents the usability of the application. E.g the UI of the project, being able to click buttons to get from point A to point B is an example of the usability of the project.

For Gamesino, the usability of the website is a big factor in retaining users. In order to meet this requirement, Gamesino's usability will meet the following requirements:

- .Gamesino will be available on Mobile phones and Computers any user can play mobile/web based games.
- The UI will be responsive.
- Users of Gamesino should be able to checkout securely fast under 30 seconds.
- Users should be able to log in and out during peak times.

- Users should be able to check their transaction history in under a minute.
- Users should be able to launch a game without delay in under a minute.
- Users should be able to withdraw money from their balance after winning an casino game in under 15 minutes.

Reliability

For Gamesino, reliability is how fast it can come back online after going offline, how fast it can recover from errors, the ability to withstand malicious activity such as XSS etc. Realibility is ensuring the application is usable after an fault or defect in the system and making it only happen for 1 out of 10 users etc.

Since Gamesino is an e-commerce gaming website, it will be impossible to delay the inevitable of server crashes, servers being overloaded during peak times, payments not going through etc, but by ensuring these issues are addressed, it will be allow for users to experience these issues once in a blue moon and allow for a better experience on Gamesino,

Performance

The performance in the FURPS represents on how well the application runs such as load times etc. Having good performance in an application is beneficial as it assures users that no time would be wasted on waiting for a singular piece of content to load. Games on Gamesino will allow users with low to high pc ranges be able to play any game on the site.

Supportability

Supportability in Gamesino refers to allowing support for Mobile Phones and Computers and making the code base compatible and easy to maintain for the two devices. Gamesino does have an support email so if issues arise user's can contact support regarding their recent purchase or any issues they have.

Metrics / Success

- Users will be able to buy/ play games.
- Users will be able to gamble with their money and able to withdraw it their bank.
- In no way would it be possible to manipulate the system to get a game for free.
- Transaction History will be able for all users
- Purchased games will be available in user's library.
- Support across Mobile and PC.
- User's Banking details are not stored
- User's personal data is encrypted.

Conclusion

This Functional Specification highlights the functional specification for gamesino. The relevant use cases are explained as well as the FURPS is also explained in regards to the application.